

# RAPPORT DU TP7

GHAMRI Samy  
Groupe TP2

GUYOLLOT Alan  
Groupe TP2

Du 24 Mars au 5 Avril 2020

**Chargé de TP :** Monsieur Giraudo

**Matière :** Perfectionnement à la programmation en C

**Etablissement / Formation :** Université Gustave Eiffel – Licence 2 Maths-info Semestre 4

# **Introduction**

L'objectif de ce TP est d'implanter le jeu du Boggle, ou plus exactement une variante dont nous préciserons les règles. Nous utilisons et approfondissons ici les notions de

- ❖ lecture d'une spécification ;
- ❖ découpage d'un projet en modules ;
- ❖ lecture efficace dans un fichier texte ;
- ❖ interface NCURSES.

Ce TP est beaucoup moins directif que les précédents. Il ressemble plus à une véritable spécification fournie par un client. Il existe ainsi plusieurs façons de mener à bien le travail demandé, mais il faut faire attention à répondre exactement à la spécification et à utiliser tout ce qui a été vu pour le moment dans les cours et TD/TP précédents (comme entre autres, la mise en place de pré-assertions, les fonctions à gestion d'erreurs, l'écriture de documentation, la bonne façon de pratiquer la modularisation et la mise en place de tests).

Une attention particulière sera également apportée à la partie documentation pour l'utilisateur du rapport, partie qui rappelons-le, explique comment installer l'application et comment l'utiliser. Il doit s'agir ici de rédiger un mode d'emploi complet recensant toutes les fonctionnalités de l'application, illustrées possiblement d'exemples et de figures.

Nous avons opté pour l'utilisation d'un **arbre lexicographique** pour représenter un dictionnaire

## **Corps du rapport**

## Modularisation du projet

### 1. Types :

- Affichage (fichier contenant toutes les fonctions d'affichage)
- Dictionnaire (fichier contenant toutes les fonctions concernant le dictionnaire)
- Grille (fichier contenant toutes les fonctions concernant la grille de Boogle)
- Include (contient tous les includes nécessaires de base)
- Mot (fichier contenant toutes les fonctions concernant les mots)

### 2. Module :

- Création d'un module pour chaque type considéré

### 3. Liste :

- Lister les fonctions principales à mettre en œuvre et les distribuer selon les modules précédents .

- Dans le module **Affichage**

- Déclarer et définir les fonctions

- - affiche\_grille
  - affiche\_score
  - affiche\_mot
  - affiche\_tentative\_restante
  - affiche\_score\_final
  - affiche\_mot\_trouve
  - affiche\_mot\_final
  - affiche\_accueil

- Dans le module **Dictionnaire**

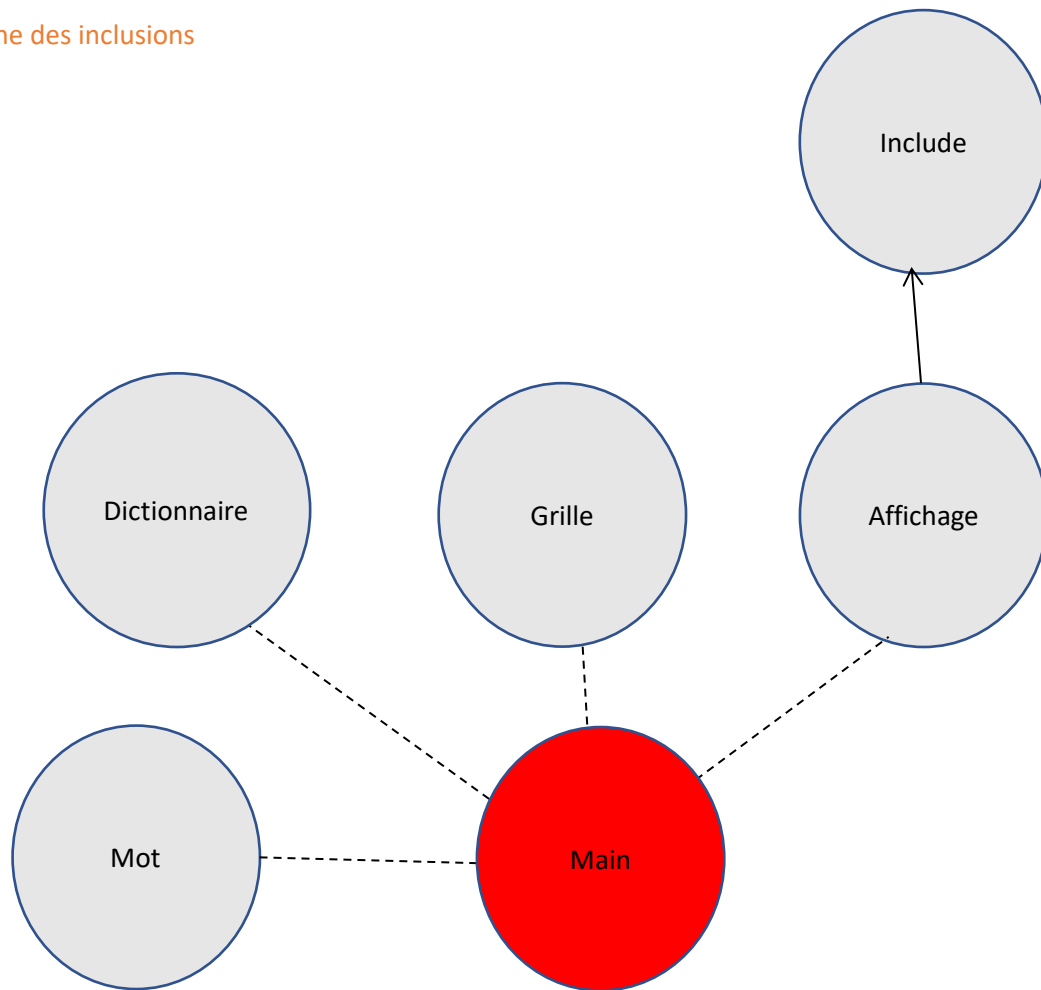
- Déclarer le type de **Dictionnaire (arbre lexicographique)**

- Déclarer et définir les fonctions

- - alloueNoeud
  - ajouteBranche
  - ajoutMot
  - recherche
  - creer\_dictionnaire

- Dans le module **Grille**
  - Déclarer de **Grille (case)**
  - Déclarer et définir les fonctions
    - `creer_grille`
    - `recherche_selectionnee`
    - `est_case_valide`
    - `reset_selectionnee`
- Dans le module **Include**
  - Contient les includes de base
- Dans le module **Mot**
  - Déclarer et définir les fonctions
    - `genere_lettre_aleatoire`
    - `longueur_mot`
    - `calcul_score`
    - `majuscule_vers_min`
    - `reset_mot`
    - `recherche_mot_trouve`

## Graphe des inclusions



## **Conclusion**



Ce TP nous a fait progressé avec l'utilisation de fichiers textes et de chaînes de caractères et notre programmation en C.

Par ailleurs, nous avons aussi pu réutiliser de nouveau des **assert()** qui sont très pratiques pour orienter les choix de l'utilisateur et éviter les erreurs du programme et modulariser notre projet afin de le découper en plusieurs petites parties et ainsi faciliter son entretien.

Nous avons pu aussi réutiliser l'interface NCURSES pour faire l'affichage de cette application.

Aucune difficulté majeure rencontrée

Aucune amélioration effectuée

# **Annexe 1**

*Instructions pour compiler le projet*

*Pour compiler le projet modularisé*

*Il suffit d'utiliser la commande **make***

# **Annexe 2**

*Documentation du projet*

### **Vous devez être sous Linux**

Tout d'abord, il faut télécharger les fichiers du projet et se déplacer où sont situés ces fichiers. Ensuite, vous ouvrez le terminal à partir de ce fichier (ou vous utilisez la commande **cd** dans le terminal pour vous déplacer dans le dossier où vous avez tout téléchargé ). Puis vous allez utiliser la commande **make** dans le terminal et pour cela vous avez juste à écrire **make** puis appuyer sur la touche **Entrée**, cela va créer le fichier exécutable. Pour lancer le jeu vous devez écrire **./Boogle** dans le terminal et appuyer sur la touche **Entrée**.

Une fois fait vous devez voir le menu du jeu s'afficher comme ceci où sont écrites les différentes règles et touches pour jouer au jeu.

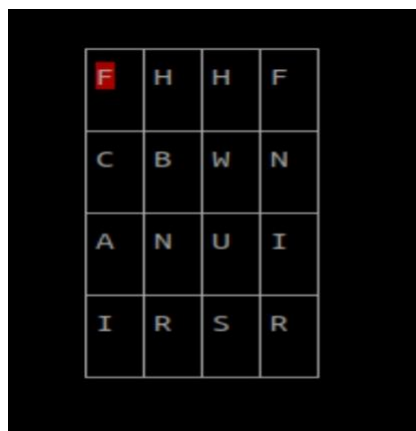
```
Bienvenue dans notre TP de BOOGLE !
Veuillez trouver ci-dessous les commandes du jeu :

-Deplacez-vous à travers la grilles grace aux flèches directionnelles
-Pour selectionner une lettre de la grille, appuyez sur espace
-Pour effacer la derniere lettre, appuyez sur A
-Pour soumettre un mot, appuyez sur ENTREE
-Vous ne pouvez pas soumettre les deux même mot
-Vous ne pouvez pas selectionner deux fois la même case pour un même mot
-Vous ne pouvez selectionner que des cases voisines

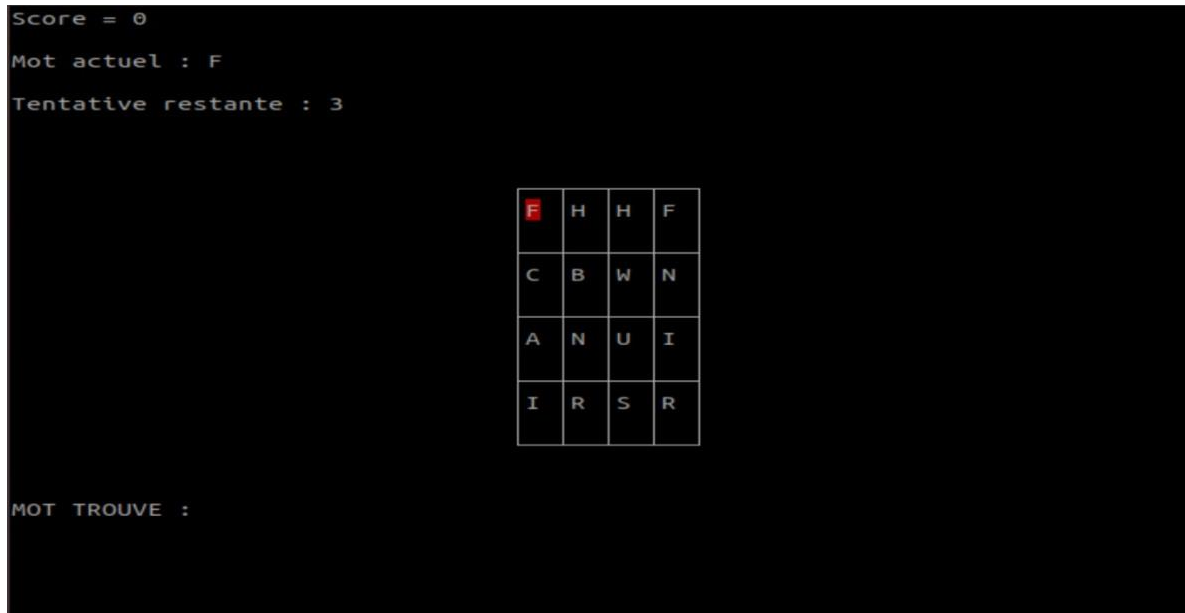
Une fois les regles lues, veuillez appuyez sur ENTREE pour demarrer !
```

Appuyez sur n'importe quelle touche pour continuer.

Utilisez les **touches directionnelles** pour déplacer le curseur rouge.

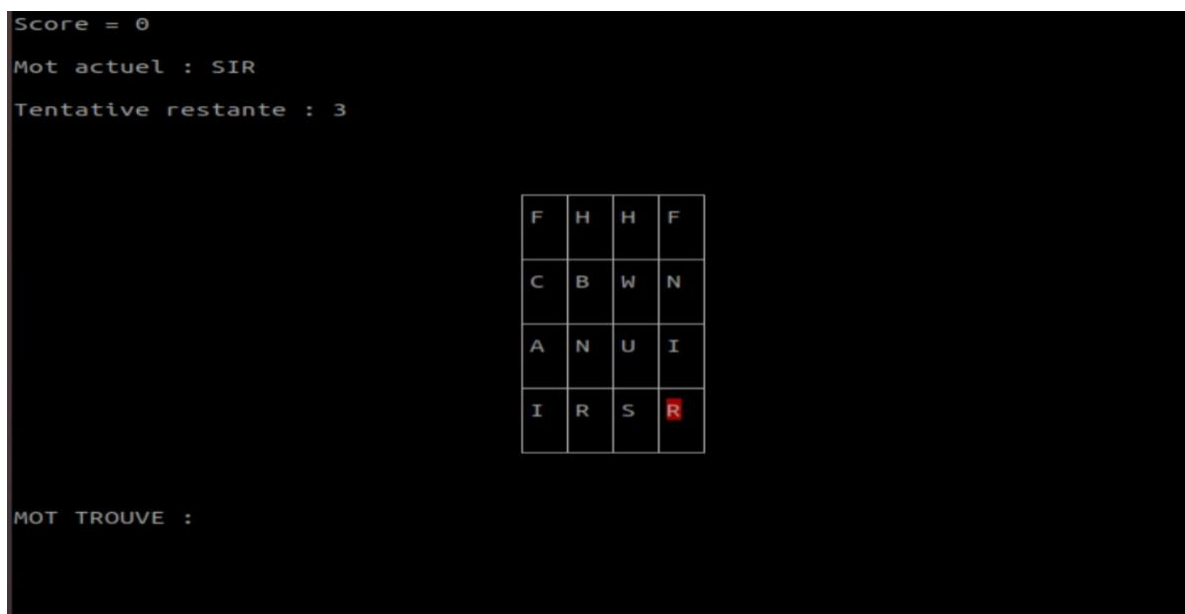


Pour sélectionner une lettre déplacer votre curseur et une fois que le curseur est sur la lettre que vous voulez prendre appuyez sur **Espace**. (Vous ne pouvez pas appuyez deux fois sur la même case pour un mot donné sinon un message s'affichera en bas à droite).



Pour la retirer appuyez sur la touche **a**.

Pour valider votre mot, appuyez sur **Entrée**, cela réduira votre nombre de tentatives restantes



Si le mot est valide, il s'affichera en bas à gauche et votre score sera actualisé.

```
Score = 1
Mot actuel :
Tentative restante : 2
```

F	H	H	F
C	B	W	N
A	N	U	I
I	R	S	R

```
MOT TROUVE :
-sir
```

Dans le cas contraire, le mot ne s'affichera pas et votre score restera inchangé.

Dès que votre nombre de tentatives atteint 0, la partie est terminée et cet écran est affiché où nous pouvons voir les bons mots trouvés qui appartiennent au dictionnaire et notre score:

```
FIN DE PARTIE !
Votre score est de : 3
Vous avez trouvé les mots suivants :
-sir
-air
```